OOD Assignment 4 Bonus Activities

These are optional bonus elements for Assignment 4. You can choose to do all, some, or none of these as part of your work and submissions for parts one and two of Assignment 4.

Bonus work for Assignment 4 Part 1 (15 points)

UML Use Case Diagram (15 points): For this diagram, imagine that the staff and visitors to the Zoo are actual people. Include the following actors in your diagram: Handler, Veterinarian, Vendor, Visitor. Show what each one of these actors could potentially do at the Zoo in terms of tasks (currently represented by the methods they invoke in the application) – you can add additional tasks not represented in the current program if you feel you need them to make the diagram more representative of interactions. Your diagram should have task options for each of the actors listed above, and you should use both optional tasks and required additional tasks shown with appropriate <includes> and <extends> links in your diagram for at least one actor. Submit the diagram as a PDF or other readable format with your Part 1 submission. Full implementation of the Use Case diagram as described above will be worth a bonus of 15 points.

Bonus work for Assignment 4 Part 2 (30 points)

JUnit test cases (15 points): For this bonus work you must create JUnit 5 test cases for your Assignment 4 Java code. You may create as many test methods as you like, but for full credit you must perform 10 assertions of some sort in the test cases (equality or object existence checks are likely the easiest to do). You must integrate JUnit into your development environment and be able to run the tests and receive feedback on how many tests passed or failed. For full credit for this bonus work you must include in your Assignment 4 Part 2 submission: required imports for JUnit modules, clearly commented code for the test cases, and a screen capture of the results of the test from your development environment (as a PDF or other easily viewable image). Full implementation of JUnit tests as described here with all deliverables will be worth 15 bonus points.

In practice, writing your tests before development is recommended, but for this academic example, we just want to demonstrated how we would add tests to Java code. If you need support on using JUnit, you will find a lot out on the web, but here are key helpful sites:

- The JUnit sites for JUnit 5 (https://junit.org/junit5/)
- Using JUnit elements in your code: https://livebook.manning.com/book/junit-recipes/chapter-3/1
- Using JUnit with IntelliJ: https://qaautomation.expert/2022/05/03/how-to-configure-junit-in-intellij/

Line graph of sales activity (15 points): For this bonus work, you are asked to create a line graph of daily sales activity at all Shops in your simulated Zoo. The line graph should include five lines of 30 datapoints for each of the five Shops where the datapoint is the daily sales $ amount (this is per day, not cumulative). You may want to add this functionality to your SalesTracker class as a method you can call with data that it is gathering over time.

Java charting libraries in common use include: JFreeChart (https://www.jfree.org/jfreechart/), charts4j (https://github.com/julienchastang/charts4j), and XChart (https://github.com/knowm/XChart).

To receive full credit for bonus points, the graph generation code in your Java source must be clear and commented, and the output for the graph must be captured as images in a PDF or other easily viewable files included in your submission.

Grading Rubric:

Assignment 4 bonus work is worth 45 points total.

I recommend you do not pursue this bonus work until you are sure the simulation itself is working well.

Part 1 (UML Use Case Diagram) is worth 15 points and is due on Friday October 13 at 11:59 PM

- The PDF or other easily viewable format for the diagram should be included with your Part 1 submission for Assignment 4.

Part 2 (JUnit tests and Line graph) is worth a total of 30 points and is due on Friday October 20 at 11:59 PM

- The code for either or both the JUnit tests and Line graph should be included with your code submission for Assignment 4. Remember to provide evidence of running the tests and/or the graph image as additional submissions for full credit.

Overall Project Guidelines

Assignments will be accepted late per the published policy from the syllabus:

- Up to two days late from posted submission date/time: -5% grade penalty
- Up to an additional three days late from posted submission date/time: -15% grade penalty
- Assignments will not be accepted after 5 days from posted submission date/time

Use office/student hours and e-mail to reach the instructor regarding homework/project questions, or if you have issues in completing the assignment for any reason.